ISEL ANO LECTIVO 22/23

PROCESSAMENTO IMAGEM E VISÃO

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Docente : João Pedro Barrigana Ramos da Costa

Aluno : Daniel João Pinheiro Infante, 41850

01

INTRODUÇÃO & OBJECTIVOS

Este trabalho foi realizado no âmbito da unidade curricular de Processamento de Imagem e Visão, da Licenciatura de Engenharia Informática e Multimédia, do Instituto Superior de Engenharia de Lisboa.

Com este trabalho pertende-se implementar de um algoritmo de visão por computador, com o objectivo de que esta consiga contar de forma automática a quantia de dinheiro, neste caso serão moedas, que se encontra sobre a mesa. A implementação deste trabalho visa ainda a familiarização com a biblioteca de OpenCV - Open Source Computer Vision - para a linguagem Python, que permite a programação de aplicações de visão por computador em tempo real.

O trabalho proposto servirá para a contagem da quantia em dinheiro, dispostas em cima de uma mesa, com uma superfície homogénia e clara, observadas por uma câmara montada em cima de um tripé, de forma a que o plano do sensor seja paralelo à mesa.

O algoritmo foi desenvolvido de modo a cumprir o seu objectivo, apesar das perturbações que possam existir no campo de visão, tais como a presença de objectos que não as moedas; existência de sombras e possível contacto com outros objectos. Para a implementação deste trabalho foi recomendado o seguimento de uma sequência típica de
operações como: leitura de imagens, conversão para níveis de cinzento, binarização, melhoramento da imagem, extracção de componentes conexos e propriedades e classificação de objectos.

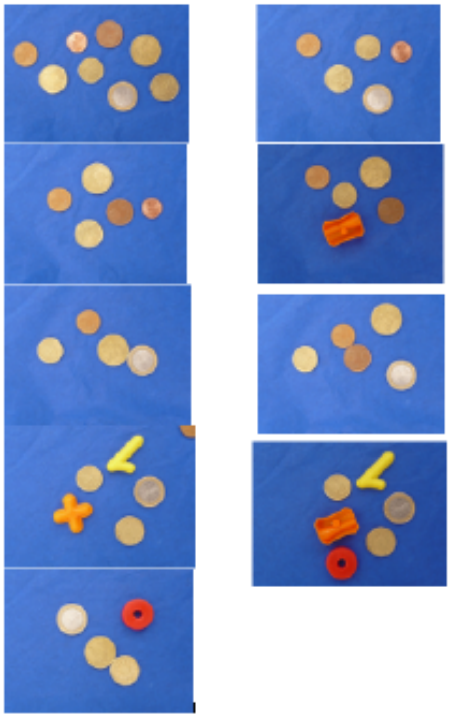
02

IMPLEMENTAÇÃO

Para a obtenção das imagens, foi utilizada uma câmara, montada em cima de um tripé para garantir que o plano do sensor era paralelo à mesma. As imagens utilizadas durante o desenvolvimento do trabalho foram fornecidas juntamente com o enunciado.

A leitura de imagens exige a utilização da função readImages, que tem como funcionalidade a leitura de um array de imagens passado no parâmetro. Nesta função é criada uma cópia deste array com vista a ser utilizado na binarização, para que as imagens iniciais não sejam danificadas.

É utilizada a função imread() do openCV, que carrega uma imagem de um local específico e retorna-a.



03 IMPLEMENTAÇÃO

Antes do processamento da imagem para tons de cinzento foi necessário realizar uma máscara que extraia os níveis de azul de modo a só ser visível os diferentes objectos em cima do fundo,

Como tal foi necessário a extração da cor, e graças ao uso de um site que permite obter o valor RGB de um dado pixel da imagem [1], foram obtidos os seguintes valores: 5

Tendo um valor base, alguma tentativa erro foi melhorado a gama de valores de modo a que a grande maioria do fundo seja eliminado, para tal foi acordado o valor de (150,65,15) para a gama dos azuis mais escuros até ao valor (240,150,100) que correspondem aos azuis mais claros.

(Nota: os valores apresentados foram convertidos para BGR para melhor coincidirem com a representação do OpenCV).

Apesar de conterem menos informação do que as imagens coloridas, as imagens de cinzento preservam os principais contornos dos objectos presentes nas imagens, os contrastes de regiões e permitem um processamento mais rápido que as imagens coloridas.
Assim, converter uma imagem para níveis de cinzento torna-se muito vantajoso.

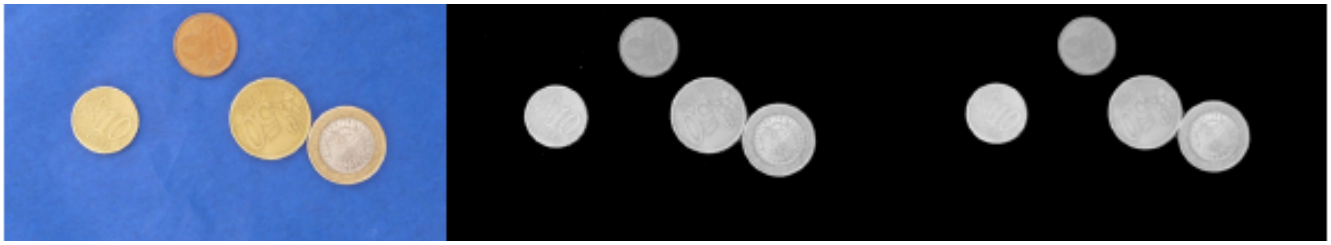
Para se realizar a conversão para níveis de cinzento foi utilizada a função mediablur do openCV, que calcula a mediana dos pixels na área do elemento estruturante(kernel) e o elemento central é substituído por esse valor médio. Esta função é altamente eficaz contra o ruído que possa existir na imagem.

Se for utilizada a função mediablur para o canal RED já não é necessário realizar a conversão explicita com função cvtColor.

04 IMPLEMENTAÇÃO

Relembrando que numa imagem no openCV em python a disposição dos planos RGB trocam para BGR, assim sendo o plano B fica na primeira posição do array das cores da imagem([0]), o G na segunda([1]) e o R na terceira ([2]).

Ao aplicar o filtro de mediana na imagem indicada, foi possível obter uma imagem ruído bastante inferior, ou seja, muito mais limpa do que se não houvesse nenhum processamento prévio.



Todas as imagens têm 4 componentes(cores). O método fornecido pelo OpenCV realiza o seguinte cálculo para obter o valor de cada pixel:

RGB[A]toGray-> Y = 0,299 * R + 0,587 * G + 0,11 * B

A binarização de imagens é utilizada quando os níveis de cinzento de uma imagem (calculados no ponto anterior), são suficientes para caracterizar os objectos.

Na binarização, um nível de cinza é considerado como um limiar de separação entre pixeis que compõem os objectos e o fundo através dela. É assim possível obter uma imagem binária, uma imagem que contém apenas dois níveis de luminância.

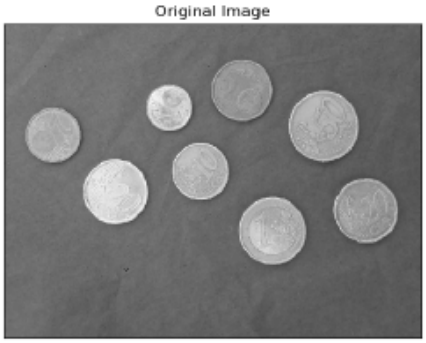
A técnica de threshold, consiste na atribuição de valores de pixeis em relação ao valor limite fornecido. No limiar, cada valor de pixel é comparado com o valor do limiar. Se o valor do pixel for menor que o limite, ele é definido como 0, caso contrário, é definido como um valor máximo.

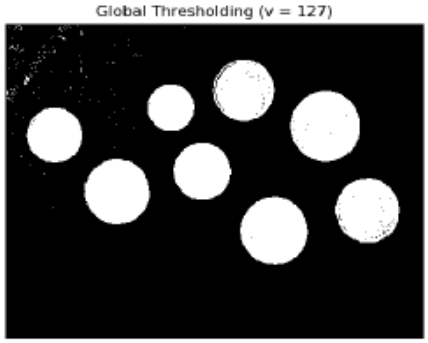
Durante o desenvolvimento foi utilizado inicialmente a função thresh_binary do openCV, que binariza a imagem com base nos valores de threshold que se insere nos parâmetros da função(valores entre 0 e 255).

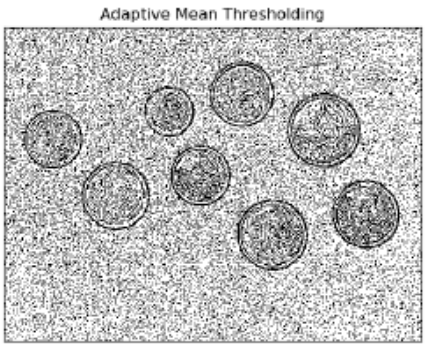
05 IMPLEMENTAÇÃO

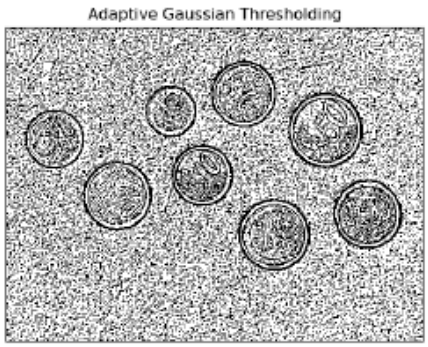
Foram testados outros métodos para binarizar a imagem no entanto apenas foi utilizada esta tecnica de threshold, pois todas as outras que experimentámos, não cumpriam os propósitos necessários.

(Nota: a imagem que foi usada abaixo não possui a filtração com máscara, no entanto para propósitos de visualização permite ser suficiente para ilustrar a razão da escolha do método):









Inicialmente foi feito um thresholding global dos pixeis com um limiar de decisão de 127 (255/2) no entanto o mesmo valor viria a trazer problemas nas moedas mais escuras, como tal foi eleito, após alguns testes, o valor de 80 para a obtenção da imagem binária.

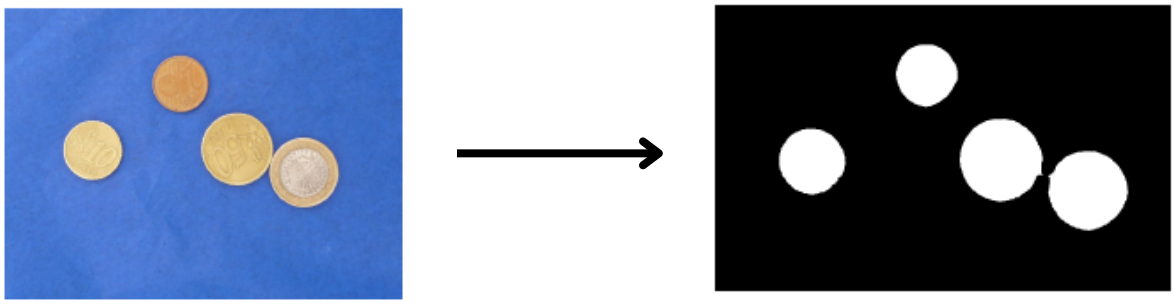
06 IMPLEMENTAÇÃO

O melhoramento da imagem é feito através de algumas experiências no processamento das imagens binárias, com o objectivo de obter os objectos diferenciados do fundo sem falhas, ou seja, todos os objectos a branco, sem pontos pretos no meio e o fundo todo preto sem pontos brancos. Para melhorar as imagens foram realizadas operações morfológicas sobre as imagens.

No desenvolvimento do algoritmo, foi calculada a matriz estruturante através da função getStructuringElement(), que foi utilizada posteriormente no cálculo do closing da imagem, que através da função morphologyEx(), aplica uma dilatação (dilate) seguida de uma erosão (erode) que replica o cálculo de uma operação de fecho, no entanto foram utilizados dois elementos estruturantes de tamanhos diferentes para uma melhor filtração.

Em seguida foi aplicada uma erosão à imagem com a função erode, com 2 iterações com uma matriz estruturante de (10,10) e por fim mais uma erosão com uma matriz de (6,6) , com vista a solucionar uma das perturbações que eras proximidade dos objectos.

Deste modo a área das moedas diminui, funcionando assim como solução para a perturbação, de modo que na fase seguinte, a extracção de componentes conexos não fosse comprometido devido à proximidade das moedas.



No âmbito do trabalho foi necessário realizar a classificação e contagem de moedas, para tal é necessário extrair os componentes conexos das imagens, de modo que forneçam ao algoritmo as informações necessárias para classificar os objectos, como por exemplo os contornos.

Os contornos unem todos os pontos contínuos que tenham a mesma intensidade ou cor. Será através dos contornos, que será possível o reconhecimento e análise de padrões/objectos.

Foram aplicadas as funções findContour() e drawContours().
A hierarquia é um vector opcional que contém informação sobre a topologia da imagem e terá tantos elementos quanto o número de contornos. Para cada índice [i] do contorno são definidos para 0 baseados em índices dos contornos do próximo e do anterior contorno.

07 IMPLEMENTAÇÃO

No mesmo nível de hierarquia, o primeiro é o contorno filho e o outro o contorno pai, respectivamente.

Se para um determinado contorno não há próximo, ou anterior, ou pai, ou contornos aninhados o correspondente elemento da hierarquia será negativo.

Através desta hierarquia de imagem que conseguimos solucionar outra das perturbações - existência de elementos com a mesma forma que as moedas - objecto com um buraco no centro. Com a aplicação da função drawCounters, desenhamos os contornos encontrados na função anterior.

Para realizar a extracção de propriedade é necessário distinguir os objectos presentes na imagem que são ou não moedas, logo é percorrido os contornos obtidos.

Foram implementados 3 formas de exclusão (tendo em conta a eliminação das perturbações descritas nos objectivos), para que, no final, sejam identificadas apenas as moedas para proceder à sua classificação e contagem.

No processo de selecção, é verificado inicialmente se algum objecto circular na imagem contém um buraco central. No caso destes elementos é verificado a hierarquia, ou seja, se existe um contorno pai e filho(se o valor não é -1), e caso exista esse mesmo contorno e o contorno pai serão descartados.

08 IMPLEMENTAÇÃO

De seguida, é verificado se os objectos são ou não circulares utilizando uma ferramenta disponibilizada pelo OpenCV que permite obter distancias entre contornos recorrendo aos cálculos de Hu Moments, que são um conjunto de 7 números calculados usando momentos centrais que são invariantes às transformações da imagem. Os primeiros 6 momentos provaram ser invariantes à translação, escala, rotação e reflexão.

Enquanto o sinal do 7º momento muda para reflexão da imagem. De seguida foi feito um calculo de diferenças entre os dois Hu moments, seguindo uma recomendação fornecida foi utilizada a seguinte formula:

$$D(A,B) = \sum_{i=0}^{6} |H_i^B - H_i^A|$$

Esta distância será calculada com base numa imagem de referência (benchmark) que foi retirada uma moeda de 1 euro, assim comparado a forma do contorno produzido pela moeda com o contorno produzido pelos restantes objectos, foi estabelecido um limiar de decisão com o valor de 0.04, este valor foi obtido numa imagem que possuía um contorno redondo no entanto encontrava-se no canto da imagem, não estando assim apto para ser classificado e logo será descartado.

09 IMPLEMENTAÇÃO

Para realizar a classificação dos objectos e obter-se o valor das moedas foi usado o algoritmo validCounters, que analisa os vizinhos.
Através deste algoritmo é calculado o valor da moeda através dos K vizinhos mais próximos. Assim, são utilizadas várias áreas em vez de apenas uma para cada moeda, de modo a tornar o cálculo e a identificação mais rigorosos. Baseado na distância, raio e valores.
Foi obtido para cada valor monetário, um valor de raio mínimo arredondando para baixo e o raio máximo em que é arredondado para cima.
Na lista seguinte, encontram-se especificados todos os raios obtidos para os diferentes valores das moedas.

• **1 centimo** 48.22298049926758 47.68654251098633 47.8984375

• **2 centimos** 57.4556884765625 57.54349899291992 57.694969177246094 56.60396957397461 57.29254150390625 57.51321029663086

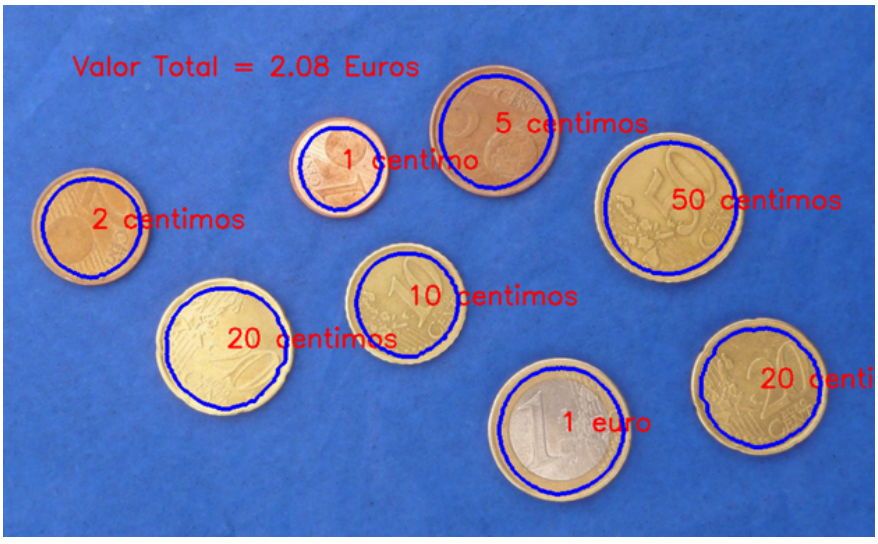
• **5 centimos** 65.6088638305664 65.39342498779297 64.03253173828125 65.30181121826172

• **10 centimos** 60.28846740722656 60.65198516845703 59.400630950927734 61.071231842041016 61.11888122558594 60.81765365600586 60.03472137451172

• **20 centimos** 69.41864776611328 70.59010314941406 69.64424896240234 69.79962921142578 69.65990447998047 68.286376953125 70.0096664428711

• **50 centimos** 76.72372436523438 77.36166381835938 76.0712661743164 78.6530532836914 76.78910827636719

• **1 euro** 73.08831787109375 73.22496795654297 75.32740020751953 73.09085083007812 72.44039154052734



10 IMPLEMENTAÇÃO

O cálculo é baseado nos 5 valores mais próximos, uma vez que é o número ímpar (permite desta forma que não haja empate de valores e dúvida em qual o valor escolhido) consideramos que é suficiente para os cálculos.

Além das moedas presentes nas imagem fornecidas adaptou-se o algoritmo para moedas de 2 euros.

Para realizar este cálculo, uma vez que não tínhamos nenhuma imagem de teste com uma moeda de 2 euros, utilizamos uma aproximação da mesma, em que como regra a moeda fornecida de maior raio é a moeda de 50 centimos, foi extrapolado que se esse raio for maior que o raio de 50 centimos, que esse contorno será de uma moeda de 2 euros.

Com o diâmetro e raio calculado, utilizamos as áreas já conhecidas será feito o calculo da diferença entre os raios de modo a extrair aquele cuja a distancia seja menor ao raio mínimo ou ao máximo de cada valor monetário.

11 CONCLUSÃO

Com o protejo realizado foi possível aplicar os conceitos sobre processamento de imagens, bem como as quais os passos necessários para a partir da imagem e da aplicação de um algoritmo sobre a mesma para poder-se extrair delas as informações necessárias, para atingir os objectivos pretendidos com este trabalho laboratorial.

Durante a implementação do algoritmo do trabalho foram encontrados alguns problemas, nomeadamente na aplicação dos operadores morfológicos, devido a junção de moedas, e ruídos das imagens.